

Application No. 09/849,804

THE CLAIMS

A detailed listing of all claims that are, or were, in the present application, irrespective of whether the claim(s) remains under examination in the application are presented below. The claims are presented in ascending order and each includes one status identifier. Those claims not cancelled or withdrawn but amended by the current amendment utilize the following notations for amendment: 1. deleted matter is shown by strikethrough for six or more characters and double brackets for five or less characters; and 2. added matter is shown by underlining.

1. (Original) A packet transformation module for editing multi-protocol streaming data packets, comprising:

an instruction memory coupled to receive search words identifying a packet type for the packet, wherein the instruction memory outputs appropriate instructions based on the packet type as indexed by the search words;

a packet memory coupled to receive one or more portions of the packet subject to editing, wherein each of the packet portions is stored in a respective memory segment of the packet memory;

a valid bit array having a plurality of memory validity fields associated with respective memory segments, wherein the state of each of the memory validity fields establishes whether the packet portion in the respective memory segment is incorporated into a resulting packet portion; and

a processing module coupled to receive the instructions from the instruction memory, and to edit the packet portions in accordance with the instructions.

Application No. 09/849,804

2. (Original) The packet transformation module as in Claim 1, further comprising a priority encoder coupled to the valid bit array to enable output of each of the packet portions associated with asserted ones of the memory validity fields.
3. (Original) The packet transformation module as in Claim 2, further comprising an output controller coupled to the priority encoder to output each of the packet portions enabled for output by the priority encoder.
4. (Original) The packet transformation module as in Claim 3, further comprising an overflow memory to receive and temporarily store second packet portions not subject to output.
5. (Original) The packet transformation module as in Claim 4, further comprising a multiplexer coupled to the output controller and the overflow memory to selectively enable transmission of the resulting packet portion from the output controller followed by the second packet portions from the overflow buffer.
6. The packet transformation module as in Claim 1, wherein the packet portions are separated by one or more blank memory segments to facilitate insertion of additional information between existing packet portions stored in the packet memory.
7. (Original) The packet transformation module as in Claim 6, wherein the resulting packet portion is read out of the packet memory in an order that the packet portions are stored in the

Application No. 09/819,804

packet memory.

8. (Original) The packet transformation module as in Claim 1, wherein the packet memory is a dual-port packet memory, and wherein the dual-port packet memory concurrently receives two portions of the packet that are subject to editing.

9. (Original) The packet transformation module as in Claim 1, further comprising a macro sequencer coupled to the packet memory to effect post-processing modifications to the packet portions in the packet memory.

10. (Original) The packet transformation module as in Claim 9, wherein the post processing modifications comprise at least one of updating a checksum, a time-to live field, a packet conformance marking, and a packet length.

11. (Original) A method for editing packets of a packet stream received at a network node, comprising:

storing one or more segments of a packet in partitionable memory segments of a modification memory;

eliciting one or more editing instructions from an instruction memory, wherein the particular editing instructions elicited is based at least in part on characteristics of the packet;

modifying at least one packet segment stored in the modification memory as directed by the editing instructions;

associating validity tags with each of the memory segments to indicate whether or

Application No. 09/E-19,804

not their corresponding packet segments will be incorporated into a resulting modified packet; and

creating the resulting modified packet by assembling the packet segments associated with the validity tags indicating incorporation into the resulting modified packet.

12. (Original) The method of Claim 11, further comprising receiving search words corresponding to the characteristics of the packet, wherein the search words are used to index the instruction memory and elicit appropriate editing instructions.

13. (Original) The method of Claim 11, wherein modifying at least one packet segment of the modification memory includes changing, adding, or deleting one or more packet segments in the modification memory.

14. (Original) The method of Claim 11, wherein modifying at least one packet segment of the modification memory comprises processing the editing instructions and modifying contents of the modification memory pursuant to the instructions.

15. (Original) The method of Claim 11, wherein modifying at least one packet segment of the modification memory comprises canceling at least one packet segment by associating an invalid tag with each of the memory segments associated with the canceled packet segment, wherein the invalid tag indicates that the packet segment will not be incorporated into the resulting modified packet.

Application No. 09/849,804

16. (Original) The method of Claim 11, wherein storing the packet segments comprises reserving memory space between the stored packet segments.
17. (Original) The method of Claim 16, wherein modifying at least a portion of the modification memory comprises inserting new packet information in the reserved memory space.
18. The method of Claim 11, further comprising interleaving available memory between the parsed portions of the packet in the modification memory, in order to allow insertion of new packet segments.
19. (Original) The method of Claim 11, wherein associating validity tags comprises setting a field of one or more bits to a first state when the packet segment is to be included in the modified packet, and setting the field to a second state when the packet segment is not to be included in the modified packet.
20. (Original) The method of Claim 11, wherein storing the packet segments in the modification memory comprises storing the packet segments in a predetermined arrangement in the modification memory.
21. (Original) The method of Claim 20, wherein creating the modified packet comprises outputting the packet segments in order of the predetermined arrangement.

Application No. 09/819,804

22. (Original) The method of Claim 11, wherein creating the modified packet comprises assembling the packet segments in ascending address order of the modification memory.
23. (Original) The method of Claim 11, wherein creating the resulting modified packet comprises assembling the packet segments in a predetermined order.
24. (Original) The method of Claim 11, further comprising determining the characteristics of the packet, including a transmission protocol associated with the packet.
25. (Original) The method of Claim 24, wherein determining the characteristics of the packet comprises parsing header information of the packet to identify the transmission protocol associated with the packet.
26. (Original) The method of Claim 11, wherein storing packet segments comprises storing parsed header information corresponding to embedded network layer headers within the packet.
27. (Original) The method of Claim 11, further comprising effecting policing modifications through direct manipulation of the packet segment in the modification memory after the packet segment has been modified pursuant to the editing instructions.
28. (Original) The method of Claim 11, further comprising serially receiving the packets from the packet stream.

Application No. 09/819,804

29. (Original) The method of Claim 11, further comprising transmitting the resulting modified packet from the network node.
30. (Original) The method of Claim 11, further comprising separating packet segments subject to modification from packet segments not subject to modification.
31. (Original) The method of Claim 30, further comprising storing the packet segments not subject to modification in an overflow buffer, wherein no modifications of the packet segments in the overflow buffer are effected.
32. (Original) The method of Claim 30, wherein the packet segments subject to modification include packet header information.
33. (Original) The method of Claim 30, wherein the packet segments not subject to modification include packet payload information.
34. (Original) The method of Claim 11, further comprising:  
storing segments of the packet which are not to be modified in an overflow memory;  
performing the modifications of the packet segments stored in the modification memory; and  
reassembling the packet to create the resulting modified packet.

Application No. 09/149,804

35. (Original) An ingress processing system for editing packets of a data stream, comprising:
- (A) a packet parser to parse each packet and generate resulting search words based on a packet protocol; and
  - (B) an editor for editing the packets, comprising:
    - (i) an instruction memory coupled to the packet parser to receive the search words, wherein the instruction memory outputs targeted instructions based on the packet protocol as indexed by the search words;
    - (ii) a packet memory coupled to receive one or more portions of the packet subject to editing, wherein each of the packet portions is stored in a respective memory segment of the packet memory;
    - (iii) a valid bit array having a plurality of memory validity fields associated with respective memory segments, wherein the state of each of the memory validity fields establishes whether the packet portion in the respective memory segment is incorporated into a resulting edited packet; and
    - (iv) a processing module coupled to receive the instructions from the instruction memory, and to edit the packet portions in accordance with the instructions.
36. (Original) A network for transferring information, comprising:
- (A) a source node to dispatch information onto the network;
  - (B) a destination node to receive the information dispatched by the source node;
  - (C) at least one intermediary node coupled along a transmission path between

Application No. 09/849,804

the source node and the destination node, wherein the intermediary node comprises:

(i) a packet parser to parse each packet and generate resulting search words based on a packet protocol; and

(ii) an editor for editing the packets, comprising:

(a) an instruction memory coupled to the packet parser to receive the search words, wherein the instruction memory outputs targeted instructions based on the packet protocol as indexed by the search words;

(b) a packet memory coupled to receive one or more portions of the packet subject to editing, wherein each of the packet portions is stored in a respective memory segment of the packet memory;

(c) a valid bit array having a plurality of memory validity fields associated with respective memory segments, wherein the state of each of the memory validity fields establishes whether the packet portion in the respective memory segment is incorporated into a resulting edited packet; and

(d) a processing module coupled to receive the instructions from the instruction memory, and to edit the packet portions in accordance with the instructions.

37. (Original) The network of Claim 36, wherein the intermediary node further comprises a policing engine coupled to the editor to provide measured packet conformance parameters to the editor to carry out post-processing modifications.

Application No. 09/849,804

38. (Original) The network of Claim 36, further comprising a switch fabric to direct the resulting edited packet to a subsequent node or the destination node.

39. (Original) A computer-readable medium having computer-executable instructions for editing multi-protocol streaming data packets, the computer-executable instructions performing steps comprising:

storing one or more segments of a packet in partitionable memory segments of a modification memory;

eliciting one or more editing instructions from an instruction memory, wherein the particular editing instructions elicited is based at least in part on characteristics of the packet;

modifying at least one packet segment stored in the modification memory as directed by the editing instructions;

associating validity tags with each of the memory segments to indicate whether or not their corresponding packet segments will be incorporated into a resulting modified packet; and

creating the resulting modified packet by assembling the packet segments associated with the validity tags indicating incorporation into the resulting modified packet.

40. (Original) A packet transformation module for editing multi-protocol streaming data packets, comprising:

means for storing one or more segments of a packet in partitionable memory

Application No. 09/849,804

segments of a modification memory;

means for eliciting one or more editing instructions from an instruction memory, wherein the particular editing instructions elicited is based at least in part on characteristics of the packet;

means for modifying at least one packet segment stored in the modification memory as directed by the editing instructions;

means for associating validity tags with each of the memory segments to indicate whether or not their corresponding packet segments will be incorporated into a resulting modified packet; and means for creating the resulting modified packet by assembling the packet segments associated with the validity tags indicating incorporation into the resulting modified packet.